K-Means+ID3: A Novel Method for Supervised Anomaly Detection by Cascading K-Means Clustering and ID3 Decision Tree Learning Methods

Shekhar R. Gaddam, Vir V. Phoha, *Senior Member*, *IEEE*, and Kiran S. Balagani

Abstract—In this paper, we present "K-Means+ID3," a method to cascade k-Means clustering and the ID3 decision tree learning methods for classifying anomalous and normal activities in a computer network, an active electronic circuit, and a mechanical mass-beam system. The k-Means clustering method first partitions the training instances into k clusters using Euclidean distance similarity. On each cluster, representing a density region of normal or anomaly instances, we build an ID3 decision tree. The decision tree on each cluster refines the decision boundaries by learning the subgroups within the cluster. To obtain a final decision on classification, the decisions of the k-Means and ID3 methods are combined using two rules: 1) the Nearest-neighbor rule and 2) the Nearest-consensus rule. We perform experiments on three data sets: 1) Network Anomaly Data (NAD), 2) Duffing Equation Data (DED), and 3) Mechanical System Data (MSD), which contain measurements from three distinct application domains of computer networks, an electronic circuit implementing a forced Duffing Equation, and a mechanical system, respectively. Results show that the detection accuracy of the K-Means+ID3 method is as high as 96.24 percent at a false-positive-rate of 0.03 percent on NAD; the total accuracy is as high as 80.01 percent on MSD and 79.9 percent on DED.

Index Terms—Anomaly detection, classification, decision trees, k-Means clustering, receiver operating characteristic (ROC) curves.

1 Introduction

Anomaly detection systems (ADS) monitor the behavior of a system and flag significant deviations from the normal activity as anomalies. Recently, anomaly detection has been used for identifying attacks in computer networks [1], malicious activities in computer systems [2], [3], [4], fatigue-cracks in mechanical systems and anomalies in electronic circuits [5], and misuses in Web systems [6], [7]. A more recent class of ADS developed using machine learning techniques like artificial neural-networks [8], Kohonen's self-organizing maps [9], fuzzy classifiers [10], symbolic dynamics [11], multivariate analysis [12], and others [13], [14], [15], [16], [17] have become popular because of their high detection accuracies at low false positive rates. However, the ADS related studies cited above have two drawbacks: 1) the majority of these works evaluate the performance of anomaly detection methods on the measurements drawn from one application domain, for example, on computer systems or networks or electronic circuits, thereby addressing the problem of anomaly detection on limited data instances collected from a single application domain, and 2) the studies build anomaly detection methods with single machine learning techniques like artificial neural-networks, pattern matching, etc., while recent advances in machine learning show that fusion [18], selection [19], and cascading [20] of multiple machine learning methods have a better performance yield over individual methods.

In this paper, we present a novel supervised anomaly detection method, called "K-Means+ID3," developed by cascading two machine learning algorithms: 1) the k-Means clustering and 2) the ID3 decision tree learning. In the first stage, k-Means clustering is performed on training instances to obtain k disjoint clusters. Each k-Means cluster represents a region of similar instances, "similar" in terms of Euclidean distances between the instances and their cluster centroids. We choose k-Means clustering because: 1) it is a data-driven method with relatively few assumptions on the distributions of the underlying data and 2) the greedy search strategy of k-Means guarantees at least a local minimum of the criterion function, thereby accelerating the convergence of clusters on large data sets. In the second stage of K-Means+ID3, the k-Means method is cascaded with the ID3 decision tree learning by building an ID3 decision tree using the instances in each k-Means cluster. Cascading the k-Means clustering method with ID3 decision tree learning alleviates two problems in k-Means clustering: 1) the *Forced Assignment* problem and 2) the *Class Dominance* problem. The *Forced Assignment* problem arises when k parameter in k-Means is set to a value that is considerably less than the inherent number of natural groupings within the training data. The k-Means procedure initialized with a low k value underestimates the natural groups within the training data and, therefore, will not capture the overlapping groups within a cluster, forcing the instances from different groups to be a part of the same cluster. Such "forced assignments" in anomaly detection may increase the false positive rate or decrease the detection accuracy. As an example of forced

- *S.R. Gaddam can be reached at gaddam@gmail.com.*
- *V.V. Phoha and K.S. Balagani are with the Department of Computer Science, Louisiana Tech University, Nethken Hall, Arizona Avenue, Ruston, LA 71270. E-mail: {phoha, ksb011}@latech.edu.*

Manuscript received 19 Sept. 2005; revised 28 June 2006; accepted 9 Oct. 2006; published online 18 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0392-0905.

assignment in an anomaly detection setting, consider an anomaly in a network traffic originating from a particular type of attack (say a "remote-to-user" attack) whose network traffic may be very similar to that of normal traffic. In this case, a low value of k parameter may force the k-Means to assign attack instances to a normal cluster because the value of k is insufficient to capture the inherent subgroup structure of the attack that differentiates it from the normal traffic, i.e., the distance (similarity) between the attack instance and the cluster representing a normal class is less than the distance between the attack instance and the cluster representing an anomaly class. The second problem, *Class Dominance* arises in a cluster when the training data have a large number of instances from one particular class and very few instances from the remaining classes. Such clusters, which are dominated by a single class, show weak association to the remaining classes. That is, when classifying an anomaly associated with a cluster dominated by normal instances or vice-versa, decisions based exclusively on the probabilistic likelihood of the instance being associated with the cluster are most likely to misclassify the instance. The *Forced Assignment* and *Class Dominance* problems cause instances from different classes, like the normal and anomaly classes in our case, to overlap within the same cluster. However, a decision tree trained on each cluster learns the subgrouping (if any) present within each cluster and refines the decision boundaries within the clusters dominated by a single class by partitioning the instances with a set of *if-then constraints* over the feature space. Cascading the decisions from the k-Means and ID3 methods involves two phases: 1) the Candidate Selection phase, and 2) the Candidate Combination phase. In the Candidate Selection phase, f clusters that are nearest in Euclidean distance between the cluster centroids and the test instance are selected. In the Candidate Combination phase, two rules are used—the Nearest-consensus rule and the Nearest-neighbor rule—to combine the decisions of the k-Means and the ID3 algorithms to obtain a final classification decision over a test instance.

We perform experiments on three data sets: 1) the Network Anomaly Data, which is feature extracted from the 1998, 1999, and 2000 MIT-DARPA network traffic [21], [22], [23] using an artificial neural network based nonlinear component analysis method presented in [24], 2) the Duffing Equation Data [5], containing measurements from an active electronic circuit implementing a forced Duffing equation, and 3) the Mechanical Systems Data [25], containing measurements drawn from a mechanical apparatus that excites a mass-beam structure for generating small fatigue cracks. The three data sets contain representative anomalous and normal behavioral patterns from three distinct domains of computer networks, an active electronic circuit system, and a mechanical system. Performance evaluation of the K-Means+ID3 cascading approach is conducted using six measures:

1. detection accuracy or true positive rate (TPR),
2. false positive rate (FPR),
3. precision,
4. total accuracy (or accuracy),
5. F-measure, and
6. receiver operating characteristic (ROC) curves and areas under ROC curves (AUCs).

The performance of K-Means+ID3 is empirically compared with the performance of individual k-Means clustering and the ID3 decision tree classification algorithms.

1.1 Contributions of the Paper

The contributions of the paper are enumerated as follows:

- The paper presents a novel method to cascade the k-Means clustering and ID3 decision tree learning methods for mitigating the *Forced Assignment* and *Class Dominance* problems of the k-Means method for classifying data originating from normal and anomalous behaviors in a computer network, an active electronic circuit, and a mechanical mass-beam system.
- The paper evaluates the performance of K-Means +ID3 classifier, and compares it with the individual k-Means clustering and ID3 decision tree methods using six performance measures.
- The paper presents a novel method for cascading two successful data partition methods for improving classification performance. From an anomaly detection perspective, the paper presents a high performance anomaly detection system.

The rest of the paper is organized as follows: In Section 2, we briefly discuss the k-Means and ID3 decision tree learning-based anomaly detection methods. In Section 3, we present the K-Means+ID3 method for anomaly detection. In Section 4, we discuss the experimental datasets. In Section 5, we discuss the results. We conclude our work and give future directions in Section 6.

2 Anomaly Detection with k-Means Clustering and ID3 Decision Tree Learning Methods

In this section, we briefly discuss the k-Means [26] clustering and the ID3 decision tree classification [27] methods for supervised anomaly detection.

2.1 Anomaly Detection with k-Means Clustering

The k-Means algorithm groups N data points into k disjoint clusters, where k is a predefined parameter. The steps in the k-Means clustering-based anomaly detection method are as follows:

1. Select k random instances from the training data subset as the centroids of the clusters $C_1, C_2, \ldots C_k$.
2. For each training instance X:
   a. Compute the Euclidean distance
   $$D(C_i, X), i = 1 \ldots k.$$
   Find cluster C_q that is closest to X.
   b. Assign X to C_q. Update the centroid of C_q. (The centroid of a cluster is the arithmetic mean of the instances in the cluster.)
3. Repeat Step 2 until the centroids of clusters $C_1, C_2, \ldots C_k$ stabilize in terms of mean-squared-error criterion.
4. For each test instance Z:
   a. Compute the Euclidean distance $D(C_i, Z)$, $i = 1 \ldots k$. Find cluster C_r that is closest to Z.
   b. Classify Z as an anomaly or a normal instance using either the *Threshold* rule or the *Bayes*

Decision rule. The *Threshold* rule for classifying a test instance Z that belongs to cluster C_r is:

$$Assign \quad Z \rightarrow 1 \quad if\ P(\omega_{1r}\,|Z \in C_r) > \tau;$$
$$Otherwise \quad Z \rightarrow 0,$$

where "0" and "1" represent normal and anomaly classes, ω_{1r} represents the anomaly class in cluster C_r, $P(\omega_{1r}|Z \in C_r)$ represents the probability of anomaly instances in C_r, and τ is a predefined threshold. In our experiments, the threshold is set to 0.5 so that a test instance is classified as an anomaly only if it belongs to a cluster that has anomaly instances in majority. The *Bayes Decision* rule is:

$$Assign \quad Z \rightarrow 1 \quad if\ P(\omega_{1r}\,|Z \in C_r)$$
$$> P(\omega_{0r}\,|Z \in C_r); \quad Otherwise \quad Z \rightarrow 0,$$

where ω_{0r} represents the normal class in cluster C_r and $P(\omega_{0r}|Z \in C_r)$ is the probability of normal instances in cluster C_r.

2.2 Anomaly Detection with ID3 Decision Trees

The ID3 decision tree learning algorithm computes the Information Gain G on each attribute A, defined as:

$$G(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v),$$

where S is the total input space and S_v is the subset of S for which attribute A has a value v. The $Entropy(S)$ over c classes is given by $\sum_{i=1}^{c} -p_i \log_2(p_i)$, where p_i represents the probability of class "i." The attribute with the highest information gain, say B, is chosen as the root node of the tree. Next, a new decision tree is recursively constructed over each value of B using the training subspace $S - \{S_B\}$. A leaf-node or a decision-node is formed when all the instances within the available training subspace are from the same class. For detecting anomalies, the ID3 decision tree outputs binary classification decision of "0" to indicate normal and "1" to indicate anomaly class assignments to test instances.

3 K-Means+ID3 Method for Anomaly Detection

We are provided with a training data set (X_i, Y_i), $i = 1, 2, \ldots, N$, where X_i represents an n-dimensional continuous valued vector and $Y_i = \{0, 1\}$ represents the corresponding class label with "0" for normal and "1" for anomaly. The K-Means+ID3 method has two steps: 1) training and 2) testing. During training, steps 1-3 of the k-Means-based anomaly detection method are first applied to partition the training space into k disjoint clusters $C_1, C_2, \ldots C_k$. Then, an ID3 decision tree is trained with the instances in each k-Means cluster. The k-Means method ensures that each training instance is associated with only one cluster. However, if there are any subgroups or overlaps within a cluster, the ID3 decision tree trained on that cluster refines the decision boundaries by partitioning the instances with a set of *if-then rules* over the feature space.

```
Input: Test instances Z_i, i = 1...n; f value.
Output: Anomaly score matrix for Z_i, i = 1...n.

Procedure Candidate_Selection {
    Step 1: For each test instance Z_i
       a. Compute Euclidean distance D(Z_i, r_j), j = 1...k,
          and find f clusters closest to Z_i.
       b. Compute k-Means and ID3 decision tree scores
          for f nearest (candidate) clusters.
    Step 2: Return Anomlay Score Matrix for Z_i.
} /* End Procedure */
```

Fig. 1. Procedure of candidate selection.

The testing step of the K-Means+ID3 has two phases: 1) the Candidate Selection phase and 2) the Candidate Combination phase. In Candidate Selection, decisions from k-Means and ID3-based anomaly detection methods are extracted. In Candidate Combination, the decisions of the k-Means and ID3 decision tree methods are combined to give a final decision on the class membership of a test instance. For combining the k-Means and ID3 decision tree methods, we present two combination rules: 1) the Nearest-neighbor rule and 2) the Nearest-consensus rule. A detailed explanation of the two phases follows.

3.1 The Candidate Selection Phase

Fig. 1 presents the procedure for the Candidate Selection. Let $DT_1, DT_2, \ldots DT_k$ be the ID3 decision trees on clusters $C_1, C_2, \ldots C_k$ formed by applying the k-Means method on the training instances. Let $r_1, r_2, \ldots r_k$ be the centroids of $C_1, C_2, \ldots C_k$, respectively. Given a test instance Z_i, the Candidate Selection procedure extracts anomaly scores from f candidate clusters $G_1, G_2, \ldots G_k$. The "f candidate clusters" are f clusters in $C_1, C_2, \ldots C_k$ that are nearest to Z_i in terms of the Euclidean distance between Z_i and the cluster centroids. Here, f is a user-defined parameter.

Fig. 2 illustrates the extraction of anomaly scores from k-Means clustering and ID3 decision tree learning methods for f candidate clusters. Let $m_1, m_2, \ldots, m_f$ represent the centroids of candidate clusters $G_1, G_2, \ldots, G_f$. Let $D(Z_i, m_1) = d_1$, $D(Z_i, m_2) = d_2$, and $D(Z_i, m_f) = d_f$, represent the Euclidean distances between the test vector Z_i and the f candidate clusters. The k-Means anomaly scores P_s, $s = 1, \ldots, f$, for each of the f candidate clusters is given by

$$P_s = P(\omega_{1s}) \times \left[1 - \frac{d_s}{\sum_{l=1}^{k} D(Z_i, r_l)}\right], \tag{1}$$

where $P(\omega_{1s})$ is the probability of anomaly instances in cluster "s." In (1), the term

$$"1 - \frac{d_s}{\sum_{l=1}^{k} D(Z_i, r_l)}"$$

is called the *Scaling Factor (SF)*. The SF scales $P(\omega_{1s})$ by weighing it against the ratio of the Euclidean distance between the cluster s and Z_i, and the sum of Euclidean distances between Z_i and the clusters $C_1, C_2, \ldots C_k$. The SF penalizes the probability of anomaly $P(\omega_{1s})$ in cluster s with its distance from the test vector Z_i, i.e., a high value of d_s

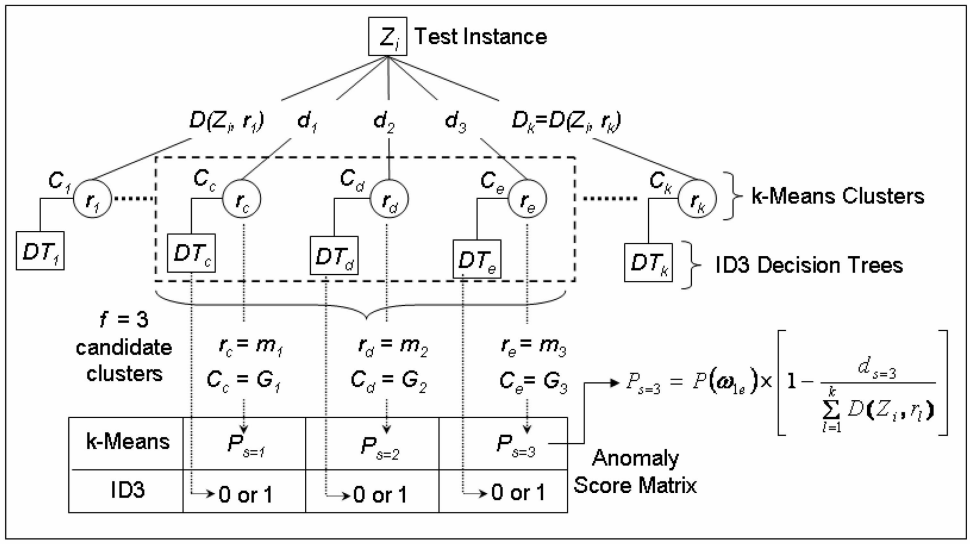

Fig. 2. Extraction of k-Means and ID3 decision tree scores from $f = 3$ candidate clusters for the test instance Z_i.

yields a low P_s value and vice versa. The decisions from the ID3 decision trees associated with the f candidate clusters are either "0" representing normal or "1" representing an anomaly classes. The Candidate Selection phase outputs an anomaly score matrix with the decisions extracted from the k-Means and ID3 anomaly detection methods for a given test vector. The decisions stored in the anomaly score matrix are combined in the Candidate Combination phase to yield a final decision on the test vector. A detailed description of the Candidate Combination follows.

3.2 The Candidate Combination Phase

The input to the Candidate Combination phase is the anomaly score matrix containing the anomaly scores P_s, $s = 1, \ldots, f$, of the k-Means and the decisions of the ID3-based anomaly detection methods over f candidate clusters. To combine the decisions of the k-Means and ID3 algorithms, we first harden the anomaly scores of the k-Means method by using the *Threshold Rule* presented in Section 2.1. Next, we use two rules: 1) the Nearest-consensus rule and 2) the Nearest-neighbor rule to combine the decisions.

3.2.1 Nearest-Consensus Rule

Fig. 3 shows an example of an anomaly score matrix for the test vector Z. The f candidate clusters $G_1, G_2, \ldots, G_f$ are ordered in the anomaly score matrix such that the distances $d_1, d_2, \ldots, d_f$ between Z and the candidate clusters $G_1, G_2, \ldots, G_f$, respectively, satisfy $d_1 < d_2 < \ldots < d_f$. In the Nearest-consensus rule, the decision of the nearest candidate cluster in which there is consensus between the decisions of the k-Means and the ID3 decision tree methods is selected as the combined classification decision. For example, in the anomaly score matrix shown in Fig. 3, the nearest consensus occurs in candidate cluster G_2 and, therefore, the test vector is classified as "1," i.e., an anomaly.

3.2.2 Nearest-Neighbor Rule

The Nearest-neighbor rule chooses the decision of the ID3 decision tree that is associated with the nearest candidate cluster within the f candidate clusters. In the anomaly score matrix shown in Fig. 3, G_1 is the nearest candidate cluster to the test vector Z. Therefore, the nearest-neighbor rule classifies the test vector as "0" (normal), which is the decision of the ID3 decision tree associated with candidate cluster G_1.

4 Data Sets

In this section, we discuss three experimental data sets: 1) Network Anomaly Data (NAD), 2) Duffing Equation Data (DED), and 3) Mechanical Systems Data (MSD). The NAD contains three data subsets: 1) NAD-98, 2) NAD-99, and 3) NAD-00, obtained by feature-extracting the 1998, 1999, and 2000 MIT-DARPA network traffic corpora [21]. The DED data set was obtained from an active nonlinear electronic circuit implementing a second-order forced Duffing equation [5]. The MSD data set [25] was obtained from an apparatus designed to induce small fatigue cracks in ductile alloy (mass-beam) structures.

Table 1 summarizes the proportion of normal and anomaly instances, and the number of dimensions in the three data sets. The training and testing data subsets were randomly drawn from the original NAD, DED, and MSD data sets. The number of instances in all the training data subsets was restricted to utmost 5,000 instances, with 70 percent of them being normal and the rest being anomaly instances. The testing data sets contain utmost 2,500 unseen (i.e., those that are not included in training data subsets)

	G_1	G_2	G_3	.....	G_f
k-Means	1	1	0	.....	1
ID3	0	1	0	.....	0

↑ Consensus

Fig. 3. An example anomaly score matrix for test instance Z. The anomaly scores of the k-Means method are hardened using the Threshold rule.

TABLE 1
Characteristics of the NAD, DED, and MSD Data Sets Used in the Anomaly Detection Experiments

Datasets		Dimensions	Training Instances		Testing Instances	
			Normal	Anomaly	Normal	Anomaly
NAD	1998	12	3500	1500	2000	500
	1999	10	3500	1500	2000	500
	2000	10	294	126	336	84
DED		4	1288	502	860	215
MSD		4	3500	1500	2000	500

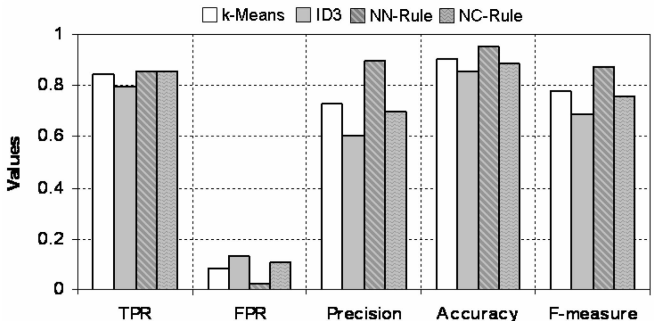

Fig. 4. Performance of the k-Means, the ID3 decision tree, and the K-Means+ID3 method with Nearest-neighbor (NN-Rule) and Nearest-consensus (NC-Rule) combination rules over the NAD-1998 test data set.

instances, with 80 percent of them being normal and the remaining 20 percent being anomaly instances. The ratio of training data sets to the testing data sets is 65 percent to 35 percent except for the NAD-2000 and DED data sets. The NAD-2000 and DED data sets contain comparatively less number of training and testing instances because of the limited number of normal instances available in DED and the limited number of anomaly instances available in NAD-2000. Therefore, the training to testing data set ratio for DED is 60 percent to 40 percent and for the NAD-2000 is 50 percent to 50 percent. A brief description of each data set follows.

4.1 Network Anomaly Data

The NAD-98, NAD-99, and NAD-00 data subsets contain artificial neural network-based nonlinear component analysis (NLCA) feature-extracted 1998, 1999, and 2000 MIT-DARPA network traffic [24], respectively. The 1998 MIT-DARPA data sets [21] were collected on an evaluation test bed simulating network traffic similar to that seen between an Air Force base (INSIDE network) and the Internet (OUTSIDE network). Thirty-eight different attacks (documented in [22]) were launched from the OUTSIDE network. Approximately seven weeks of training data and two weeks of test data were collected by a sniffer deployed between the INSIDE and OUTSIDE network. List files provide attack labels for the seven-week training data. However, the list files associated with the test data do not contain attack labels. For this reason, we use only the seven-week training data for both training and testing purposes. The 1999 MIT-DARPA data sets [23] were generated on a test bed similar to that used for 1998 MIT-DARPA data sets. Twenty-nine additional attacks (documented in [23]) were developed. The data sets contain approximately three weeks of training data (with two weeks of data exclusively containing normal traffic) and two weeks of test data. In our experiments, we use the tcpdumps collected by the sniffer in the INSIDE network on weeks 1, 3, 4, and 5. The tcpdumps from Week-2 were excluded because the list files associated with data sets were not available. The 2000 MIT-DARPA data sets [23] are attack-scenario specific data sets. The data sets contain three attack scenarios simulated with the background traffic being similar to those in 1999 MIT-DARPA data sets. The first data set, LLS DDOS 1.0, simulates a 3.5 hour attack scenario in which a novice attacker launches a Distributed Denial of Service (DDoS) attack against a naive adversary. The second data set, LLS DDOS 2.0.2, is a two-hour stealthy DDoS attack scenario. The third data set, Windows NT Attack, is a nine-hour data set containing five phased Denial-of-Service (DoS) attack on Windows NT hosts.

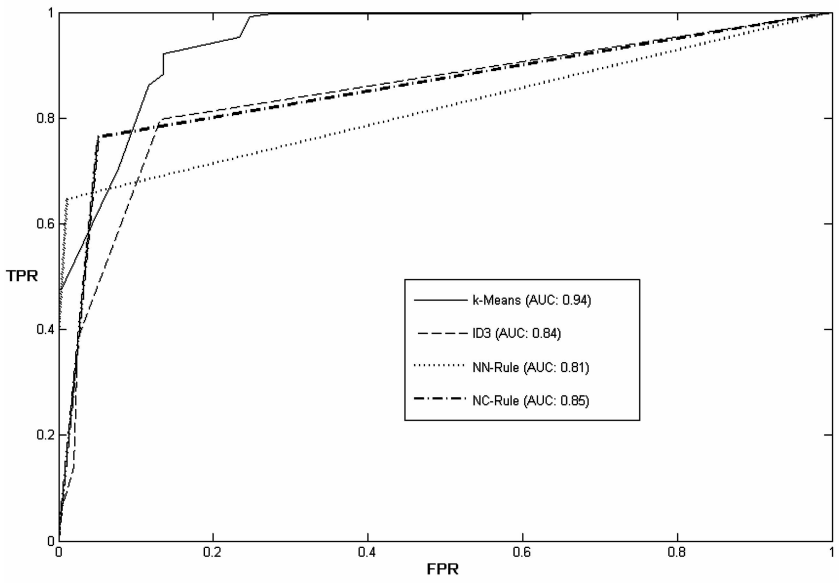

Fig. 5. ROC Curves and AUCs of k-Means, ID3, and K-Means+ID3 with NN-Rule and NC-Rule over the NAD-1998 test data set.

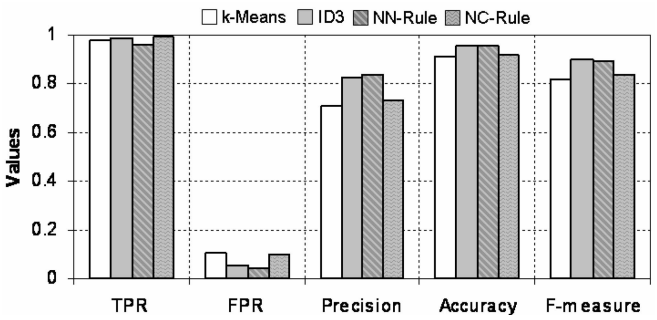

Fig. 6. Performance of the k-Means, the ID3 decision tree, and the K-Means+ID3 method with Nearest-neighbor (NN-Rule) and Nearest-consensus (NC-Rule) combination rules over the NAD-1999 test data set.

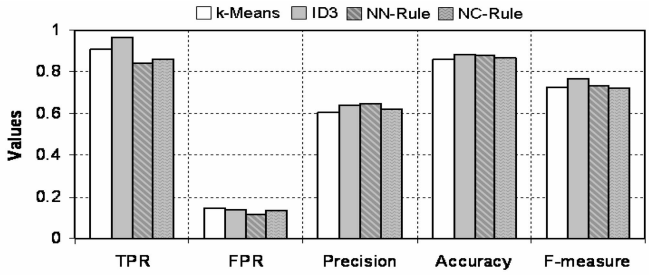

Fig. 8. Performance of the k-Means, the ID3 decision tree, and the K-Means+ID3 method with Nearest-neighbor (NN-Rule) and Nearest-consensus (NC-Rule) combination rules over the NAD-2000 test data set.

4.2 Duffing Equation Data

This section describes the preparation of Duffing Equation Data set (DED). Chin et al. [5] use an active nonlinear electronic circuit to generate the data. The circuit implements a second-order, nonautonomous, forced Duffing equation represented as:

$$\frac{\partial^2 x(t)}{\partial t^2} + \beta(t_s)\frac{\partial x(t)}{\partial t} + x(t) + x^3(t) = A\,\mathbf{cos}\,\omega t. \qquad (2)$$

The dissipation parameter $\beta(t_s)$, implemented as resistance in the circuit, varies in the slow-time t_s and is constant in the fast time-scale t at which the dynamical system is excited. Although the system dynamics is represented by a low order differential equation, it exhibits chaotic behavior that is sufficiently complex from thermodynamic perspectives and is adequate for illustration of the anomaly detection concept. The goal is to detect the changes in $\beta(t_s)$, which are associated with an anomaly. Setting the stimulus with amplitude $\mathrm{A} = 5.5$ and $\omega = 5.0\ \mathrm{rad/sec}$, the stationary behavior of the system response for this input stimulus is obtained for several values of β in the range of 0.10 to 0.35. In all our experiments with DED, we have considered the data sets with $\beta = 0.1$, $\beta = 0.32$, $\beta = 0.33$, $\beta = 0.34$, and $\beta = 0.35$ to randomly select 1,790 instances for preparing the training data subsets and 1,075 unseen random instances for preparing the test data subset.

4.3 Mechanical System Data

This section discusses the preparation of Mechanical System Data (MSD). Khatkhate et al. [25] present the test apparatus to generate the MSD. The test apparatus has two subsystems: 1) the plant subsystem consisting of the mechanical structure including test specimens (i.e., the mass-beams that undergo fatigue crack damage), electro-magnetic shakers, and displacement measuring sensors; and 2) the instrumentation and control subsystem consisting of the hardware and software components related to data acquisition and processing. The mechanical structure of the test apparatus is persistently excited near resonance to induce a stress level that causes fatigue cracks in the mass-beam specimens and yields an average life of approximately 20,000 cycles or 36 minutes. The mass-beams attain stationary behavior in the fast-time scale of machine vibrations when persistently excited in the vicinity of its resonant frequency. Fatigue cracks occur at a slow time scale that is slow relative to the fast time scale dynamics of the vibratory motion. The goal here is to detect the slowly

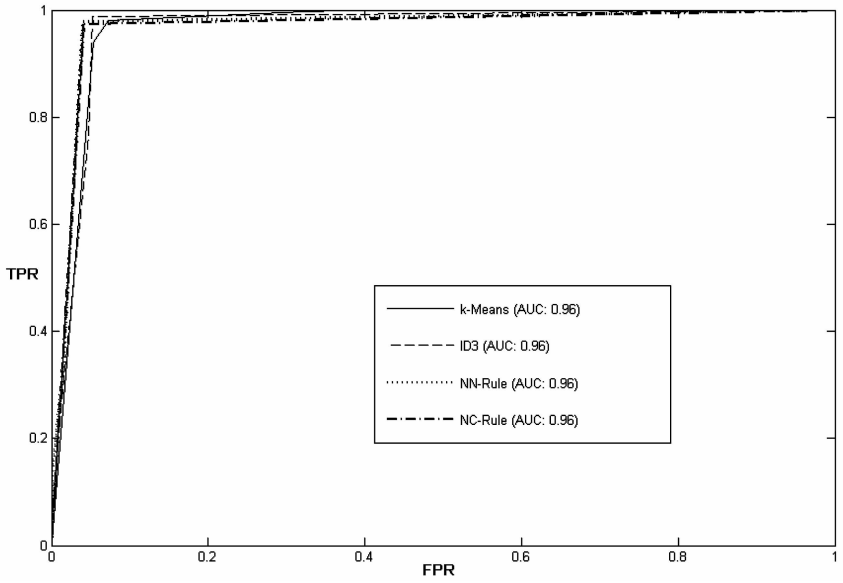

Fig. 7. ROC Curves and AUCs of k-Means, ID3, and K-Means+ID3 with NN-Rule and NC-Rule over the NAD-1999 test data set.

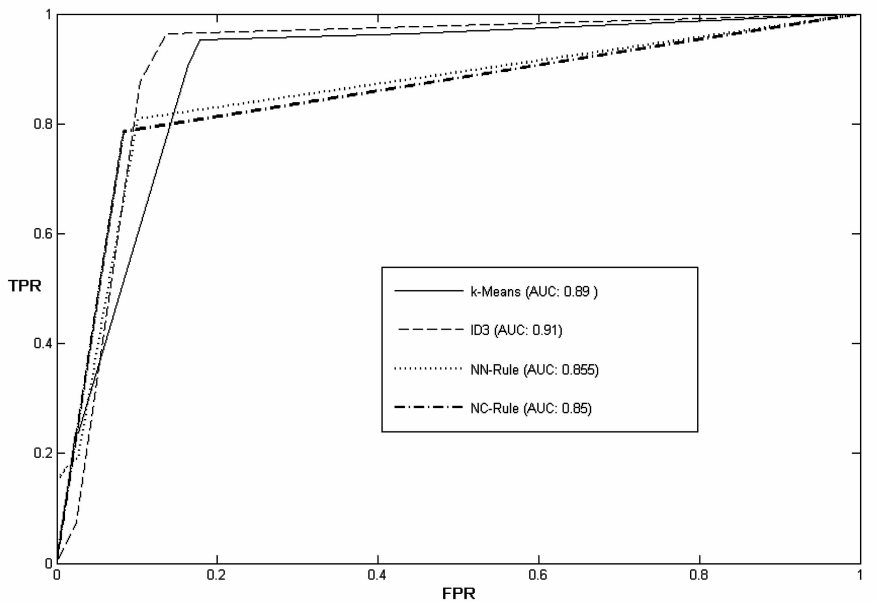

Fig. 9. ROC Curves and AUCs of k-Means, ID3, and K-Means+ID3 methods over the NAD-2000 test data set.

evolving anomaly, possibly due to fatigue cracks, by observing the time series data from displacement measuring sensors. There is a total of 36 minutes of data. The first two minutes of data is considered to be transient (normal) and the rest from 3 to 36 minutes of data is considered as steady state asymptotic behavior (anomaly). In all our experiments with MSD, we use the data recorded during the first, 33rd, 34th, 35th, and the 36th minute to randomly select 5,000 instances for preparing the training data subsets and 2,500 unseen random instances for preparing the test data subset.

5 Results and Discussion

In this section, we present the results of the K-Means+ID3 method with the Nearest-neighbor and Nearest-consensus combination rules and compare it with the individual k-Means and ID3 decision tree methods over the NAD, DED, and MSD data sets. We use six measures for comparing the performance:

1. TPR or recall is the percentage of anomaly instances correctly detected,
2. FPR is the percentage of normal instances incorrectly classified as anomaly,
3. "precision" is the percentage of correctly detected anomaly instances over all the detected anomaly instances,
4. "total accuracy" or "accuracy" is the percentage of all normal and anomaly instances that are correctly classified,
5. the "F-measure" is the equally-weighted (harmonic) mean of precision and recall, and
6. the ROCs [27] and AUCs [28] give the performance of an anomaly detection system with FPR on the x-axis and TPR on the y-axis.

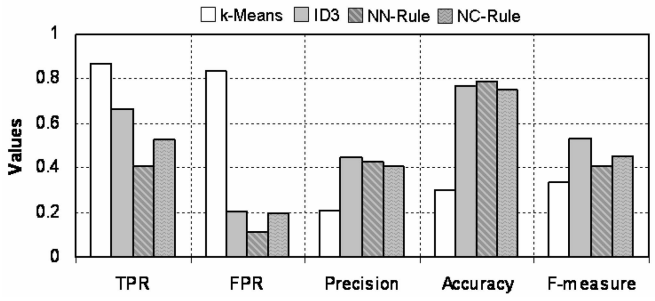

Fig. 10. Performance of the k-Means, the ID3 decision tree, and the K-Means+ID3 method with Nearest-neighbor (NN-Rule) and Nearest-consensus (NC-Rule) combination rules over the DED test data set.

The performance measures precision, recall, and F-measure determine how the K-Means+ID3, the k-Means, and the ID3 methods perform in identifying anomaly instances. The performance measure "accuracy" determines the number of normal and anomaly instances correctly classified by the anomaly detection methods. The measures FPR and AUC determine the number of false positives that the anomaly detection systems generate at specific detection accuracies. The results of our experiments on the NAD, DED, and MSD follow.

5.1 Results on the NAD-1998 Data Set

Here, we present the results of the k-Means and ID3 decision tree-based anomaly detection methods and the K-Means+ID3 method over the NAD-1998 data sets.

Fig. 4 illustrates the performance of the k-Means, the ID3, and the K-Means+ID3 methods averaged over 10 trials for k-means and K-means+ID3. For the NAD-1998 data sets, the k value of the k-Means method was set to 20. For the ID3, the training space was discretized into 45 equal-width intervals. For the K-Means+ID3 cascading method the k was set to 20 and the data was discretized into 45 equal-width intervals. The choice of k value used in our experiments was based on 10 trial experiments conducted with k set to 5, 10, 15, and 20. The performance of the k-Means-based anomaly detection did not show any significant improvement when k value was set to a value greater than 20. Similarly, the choice of the number of equal-width intervals for discretization was based on 19 experiments conducted with

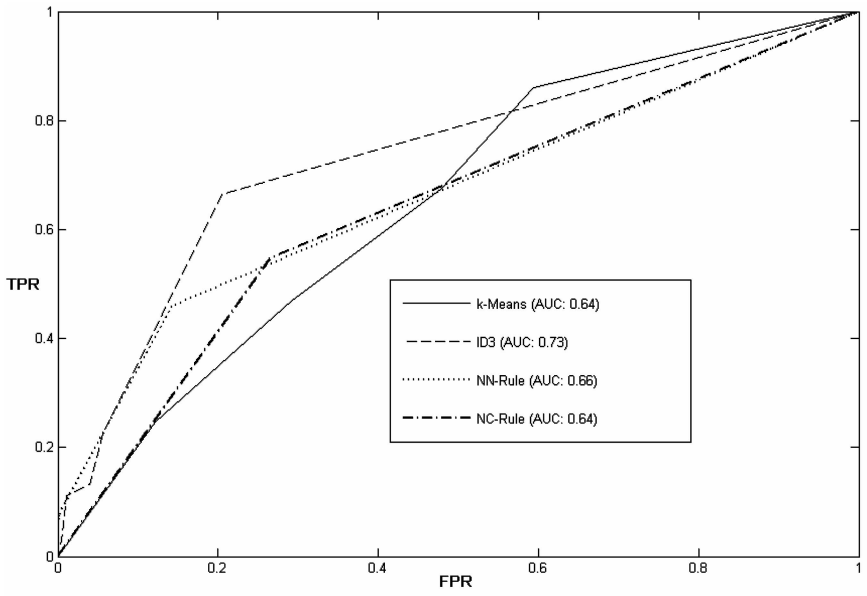

Fig. 11. ROC Curves and AUCs of k-Means, ID3, and K-Means+ID3 methods over the DED test data set.

different discretization values (e.g. 10, 15, ..., 100). Fig. 4 shows that: 1) the K-Means+ID3 cascading method based on Nearest-neighbor (NN) combination rules has better performance than the k-means and ID3 in terms of TPR, FPR, Precision, and Accuracy, 2) the TPR, FPR, Precision, Accuracy, and F-measure of the K-Means+ID3 cascading with NC combination is in-between the k-Means and the ID3, and 3) the K-Means+ID3 with NN combination outperforms the k-Means and ID3 algorithms in terms of F-measure, obtained from combining precision and recall.

Fig. 5 shows the ROC curves and AUC values for the k-Means, ID3, and K-Means+ID3 methods. The ROC curves for the K-Means+ID3 and the k-Means algorithms were plotted for the trials with the AUC values that are closest to the mean TPR values shown in Fig. 4. The ROC for K-Means+ID3 cascading algorithm with NN combination rule shows that the best TPR is achieved at 0.76 with an FPR as low as 0.05.

5.2 Results on the NAD-1999 Data Set

Fig. 6 illustrates the performance of the k-Means, the ID3, and the K-Means+ID3 methods averaged over 10 trials for k-Means and K-Means+ID3. For the NAD-1999 data sets, the k value of individual k-Means was set to 5. For the ID3 algorithm, the training space was discretized into 25 equal-width intervals. For the K-Means+ID3 cascading, the value of k was set to 5 and the data was discretized into 25 equal-width intervals. Fig. 6 shows that: 1) the K-Means+ID3 cascading with NC combination has better performance than the k-Means and ID3 in terms of TPR, and 2) precision, accuracy, and F-measure of the K-Means+ID3 with NN combination is higher than the k-Means and ID3.

Fig. 7 shows the ROC curves and AUC values of the k-Means, ID3 and K-Means+ID3 methods over NAD-1999. The ROC curves for K-Means+ID3 and k-Means method were plotted for the trial with the AUC values closest to the mean TPR values shown in Fig. 6. The K-Means+ID3 cascading with NN and NC combination has the same AUC performance as compared to k-Means and ID3 methods.

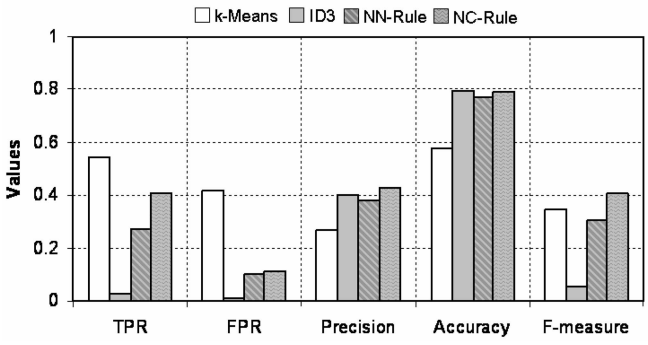

Fig. 12. Performance of the k-Means, the ID3 decision tree, and the K-Means+ID3 method with Nearest-neighbor (NN-Rule) and Nearest-consensus (NC-Rule) combination rules over the MSD test data set.

5.3 Results on the NAD-2000 Data Set

Fig. 8 illustrates the performance of the k-Means, the ID3, and the K-Means+ID3 methods averaged over 10 trials for k-Means and K-Means+ID3. For the NAD-2000 data sets, the k value of the k-Means was set to 10. For the ID3 algorithm, the training space was discretized into 15 equal-width intervals. For the K-Means+ID3 cascading algorithm, we set the value of k to 10 and discretized the data into 15 equal-width intervals. Fig. 8 shows that: 1) the K-Means+ID3 cascading with NN combination has better performance than the k-Means and ID3 in terms of FPR and Precision, 2) the TPR of the K-Means+ID3 cascading is less than the k-Means and ID3 methods, and 3) the accuracy of the K-Means+ID3 is similar to the k-Means and ID3 methods.

Fig. 9 shows the ROC curves and AUC values of the k-Means, ID3, and K-Means+ID3 methods over NAD-2000 test data set. The ROC curves for the K-Means+ID3 and k-Means methods were plotted for the trial with the AUC value that is closest to the mean TPR values in Fig. 8. The ROC curves for the k-Means and ID3 methods show better performance than the K-Means+ID3 cascading algorithm over the NAD-2000 data sets.

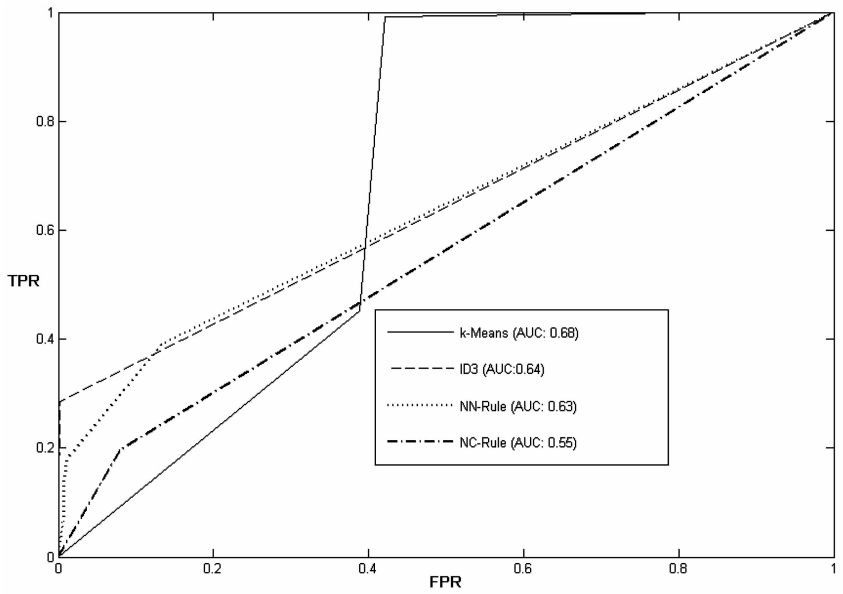

Fig. 13. ROC Curves and AUCs of k-Means, ID3, and K-Means+ID3 methods over the MSD test data set.

5.4 Results on the DED Data Set

Fig. 10 illustrates the performance of the k-Means, the ID3, and the K-Means+ID3 methods averaged over 10 trials for k-Means and K-Means+ID3. For the DED data sets, the k value for the k-Means was set to five clusters. For the ID3, the training space was discretized into 45 equal-width intervals. For the K-Means+ID3 method, we set the value of k to 5 and discretized the data into 45 equal-width intervals. Fig. 10 shows that: 1) the K-Means+ID3 cascading with NC and NN combination has better performance than the k-Means in terms of FPR, precision, and accuracy, 2) the F-measure of the K-Means+ID3 cascading is in-between the k-Means and the ID3, and 3) the TPR of the k-Means+ID3 is less than the k-Means and ID3 methods.

Fig. 11 shows the ROC curves and AUC values of the k-Means, ID3 and K-Means+ID3 methods over DED. The ROC curves for K-Means+ID3 and k-Means algorithm were plotted for the trial with the AUC value that is closest to the mean TPR values shown in Fig. 10. The ROC curve for the K-Means+ID3 cascading with NC and NN combinations is in-between the k-Means and the ID3 methods over the DED test data sets.

5.5 Results on the MSD Data Set

Fig. 12 illustrates the performance of the k-Means, the ID3, and the K-Means+ID3 algorithms averaged over 10 trials for k-Means and K-Means+ID3. For the MSD data sets, the k value of the k-Means was set to 5. For the ID3 method, the training space was discretized into 65 equal-width intervals. For the K-Means+ID3 method, we set the value of k to 5 and discretize the data into 65 equal-width intervals. Fig. 12 shows that: 1) K-Means+ID3 with NC combination has better performance than the k-Means in terms of FPR, precision, and F-measure, and 2) the precision, accuracy, and the F-measure of the K-Means+ID3 with NC combination is higher than the k-Means method.

Fig. 13 shows the ROC curves and AUCs of the k-Means, ID3, and K-Means+ID3 methods over DED. The ROC curves for K-Means+ID3 and k-Means methods were plotted for the trial with the AUC value that is closest to the mean TPR values in Fig. 12. The ROC curves for the K-Means+ID3 with NN combination shows a TPR rate as high as 0.98 at a FPR of 0.4 over the MSD test data set.

6 Conclusion and Future Work

In this paper, we developed the K-Means+ID3 pattern recognition method for anomaly detection. The K-Means +ID3 method is based on cascading two well-known machine learning methods: 1) the k-Means and 2) the ID3 decision trees. The k-Means method is first applied to partition the training instances into k disjoint clusters. The ID3 decision tree built on each cluster learns the subgroups within the cluster and partitions the decision space into finer classification regions; thereby improving the overall classification performance. We compare our cascading method with the individual k-Means and ID3 methods in terms of the overall classification performance defined over six different performance measures. Results on the NAD, DED, and MSD data sets show that:

1. the K-Means+ID3 method outperforms the individual k-Means and the ID3 in terms of all the six performance measures over the NAD-1998 data sets,
2. the K-Means+ID3 method has a very high detection accuracy (99.12 percent) and AUC performance (0.96) over the NAD-1999 data sets,
3. the K-Means+ID3 method shows better FPR and precision performance as compared to the k-Means and ID3 over the NAD-2000,
4. the FPR, Precision, and the F-measure of the K-Means+ID3 is higher than the k-Means method and lower than the ID3 methods over the NAD, and
5. the K-Means+ID3 method has the highest Precision and F-measure values over the MSD.

Future directions in this research include: 1) developing theoretical error bounds for the K-Means+ID3 method, and 2) comparing the performance of K-Means+ID3 with cascading classifiers developed using different clustering methods like hierarchical clustering, adaptive resonance

(ART) neural networks, and Kohonen's self-organizing maps and decision trees like C4.5 and Classification and Regression Trees (CART).

ACKNOWLEDGMENTS

This work was supported in part by the US Army Research Office under Grant No. DAAD 19-01-1-0646. The authors thank Dr. Asok Ray, Pennsylvania State University, for providing the Duffing Equation Data Set and the Mechanical System Data Set.

REFERENCES

[1] A. Lazarevic, A. Ozgur, L. Ertoz, J. Srivastava, and V. Kumar, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," *Proc. SIAM Int'l Conf. Data Mining,* May 2003.

[2] N. Ye, Y. Zhang, and C.M. Borror, "Robustness of the Markov-Chain Model for Cyber-Attack Detection," *IEEE Trans. Reliability,* vol. 53, no. 1, pp. 116-123, 2004.

[3] D. Mutz, F. Valeur, G. Vigna, and C. Kruegel, "Anomalous System Call Detection," *ACM Trans. Information and System Security,* vol. 9, no. 1, pp. 61-93, Feb. 2006.

[4] S. Kumar and E.H. Spafford, "A Pattern Matching Model for Misuse Intrusion Detection," *Proc. 17th Nat'l Computer Security Conf.,* pp. 11-21, Oct. 1994.

[5] S.C. Chin, A. Ray, and V. Rajagopalan, "Symbolic Time Series Analysis for Anomaly Detection: A Comparative Evaluation," *Signal Processing,* vol. 85, no. 9, pp. 1859-1868, Sept. 2005.

[6] M. Thottan and C. Ji, "Anomaly Detection in IP Networks," *IEEE Trans. Signal Processing,* vol. 51, no. 8, pp. 2191-2204, 2003.

[7] C. Kruegel and G. Vigna, "Anomaly Detection of Web-Based Attacks," *Proc. ACM Conf. Computer and Comm. Security,* Oct. 2003.

[8] Z. Zhang, J. Li, C.N. Manikopoulos, J. Jorgenson, and J. Ucles, "HIDE: A Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification," *Proc. 2001 IEEE Workshop Information Assurance,* pp. 85-90, June 2001.

[9] S.T. Sarasamma, Q.A. Zhu, and J. Huff, "Hierarchical Kohonen Net for Anomaly Detection in Network Security," *IEEE Trans. Systems, Man, and Cybernetics-Part B,* vol. 35, no. 2, Apr. 2005.

[10] J. Gomez and D.D. Gupta, "Evolving Fuzzy Classifiers for Intrusion Detection," *Proc. 2002 IEEE Workshop Information Assurance,* June 2001.

[11] A. Ray, "Symbolic Dynamic Analysis of Complex Systems for Anomaly Detection," *Signal Processing,* vol. 84, no. 7, pp. 1115-1130, 2004.

[12] N. Ye, S.M. Emran, Q. Chen, and S. Vilbert, "Multivariate Statistical Analysis of Audit Trails for Host-Based Intrusion Detection," *IEEE Trans. Computers,* vol. 51, no. 7, pp. 810-820, 2002.

[13] H.S. Javitz and A. Valdes, "The SRI IDES Statistical Anomaly Detector," *Proc. IEEE Symp. Security and Privacy,* pp. 316-326, May 1991.

[14] I. Levin, "KDD-99 Classifier Learning Contest: LLSoft"s Results Overview," *SIGKDD Explorations,* vol. 1, pp. 67-75, Jan. 2000.

[15] D.Y. Yeung and C. Chow, "Parzen-Window Network Intrusion Detectors," *Proc. 16th Int'l Conf. Pattern Recognition,* vol. 4, pp. 385-388, Aug. 2002.

[16] R. Agarwal and M.V. Joshi, "PNrule: A New Framework for Learning Classifier Models in Data Mining (A Case-Study in Network Intrusion Detection)," Technical Report DSTO-GD-0286, Dept. of Computer Science, Univ. of Minnesota, 2000.

[17] G. Qu, S. Hariri, and M. Yousif, "A New Dependency and Correlation Analysis for Features," *IEEE Trans. Knowledge and Data Eng.,* vol. 17, no. 9, pp. 1199-1207, Sept. 2005.

[18] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, "On Combining Classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 3, pp. 226-239, Mar. 1998.

[19] A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, and A. Gelzinis, "Soft Combination of Neural Classifiers: A Comparative Study," *Pattern Recognition Letters,* vol. 20, pp. 429-444, 1999.

[20] L.I. Kuncheva, "Switching between Selection and Fusion in Combining Classifiers: An Experiment," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 32, no. 2, pp. 146-156, Apr. 2002.

[21] R.P. Lippman, D.J. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R.K. Cunningham, and M.A. Zissman, "Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation," *Proc. DARPA Information Survivability Conf. and Exposition (DISCEX '00),* pp. 12-26, Jan. 2000.

[22] J. Haines, L. Rossey, R.P. Lippman, and R.K. Cunningham, "Extending the DARPA Offline Intrusion Detection Evaluation," *Proc. DARPA Information Survivability Conf. and Exposition (DISCEX '01),* June 2001.

[23] R. Lippmann, J.W. Haines, D.J. Fried, J. Korba, and K. Das, "The 1999 DARPA Off-Line Intrusion Detection Evaluation," *Proc. Third Int'l Workshop Recent Advances in Intrusion Detection (RAID '00),* pp. 162-182, Oct. 2000.

[24] G.K. Kuchimanachi, V.V. Phoha, K.S. Balagani, and S.R. Gaddam, "Dimension Reduction Using Feature Extraction Methods for Real-Time Misuse Detection Systems," *Proc. IEEE 2004 Information Assurance Workshop,* pp. 195-202, June 2004.

[25] A.M. Khatkhate, A. Ray, E. Keller, and S. Chin, "Symbolic Time Series Analysis of Mechanical Systems for Anomaly Detection," *IEEE/ASME Trans. Mechatronics,* vol. 11, no. 4, pp. 439-447, Aug. 2006.

[26] R. Duda, P. Hart, and D. Stork, *Pattern Classification,* second ed. Wiley Publishers, Oct. 2000.

[27] T. Mitchell, *Machine Learning.* McGraw-Hill, 1997.

[28] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Data Mining Researchers," Technical Report HPL-2003-4, HP Labs, 2003.

[29] J. Huang and C. Ling, "Using AUC and Accuracy in Evaluating Learning Algorithms," *IEEE Trans. Knowledge and Data Eng,* vol. 17, no. 3, pp. 299-310, Mar. 2005.

Shekhar R. Gaddam received the MS degree in computer science from Louisiana Tech University. He is currently working as a software consultant and is developing secure, robust, and scalable Web portals for managing financial transactions. His research interests include anomaly detection in computer networks and Web security.

Vir V. Phoha (M'96-SM'03) received the MS and PhD degrees in computer science from Texas Tech University, Lubbock. He is a professor of computer science at Louisiana Tech University, Ruston. His research interests include anomaly detection, network and Internet security, Web mining, control of software systems, intelligent networks and nonlinear systems. Dr. Phoha is a member of the ACM and a senior member of the IEEE.

Kiran S. Balagani received the MS degree in computer science from Louisiana Tech University. He is a research assistant in Louisiana Tech's Anomaly Detection and Mitigation Laboratory and is working toward the PhD degree in computational analysis and modeling. His research interests are anomaly detection in computer systems and networks, misuse detection, machine learning, artificial neural networks, and Web mining.